

データマイニングと情報可視化

Week 1

稲垣 紫緒

いながき しお

理学研究院 物理学部門 / 共創学部

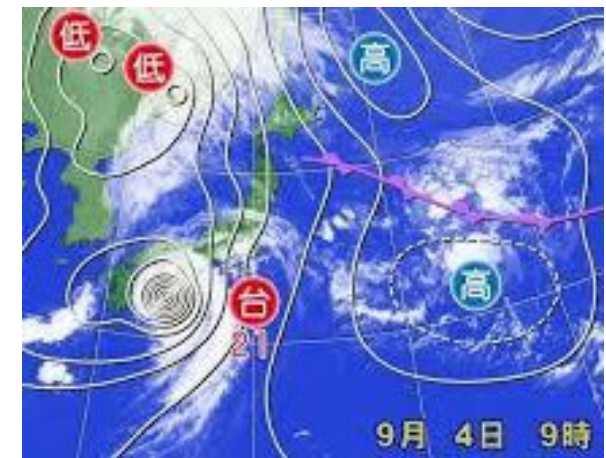
inagaki@phys.kyushu-u.ac.jp

ウェスト1号館 W1-A823号室

身近にあるデータ

オンラインショッピングや購買履歴
インターネットバンキング
ポイントカードを使った、コンビニなどの購買履歴
アンケート

天気予報
交通渋滞



データマイニングでできること



① データの分類

② データの関連性を見つけ出す

③ 事象の発生確率を予測する

データは基本的に数字や文字の羅列なので、
結果を解釈しやすい形に**可視化**する必要がある



マネーボール(2011)



○従来の野球スカウト
⇒経験と勘!!

○ビリー・ビーン

アスレックスGM

ポール・デポDESTA

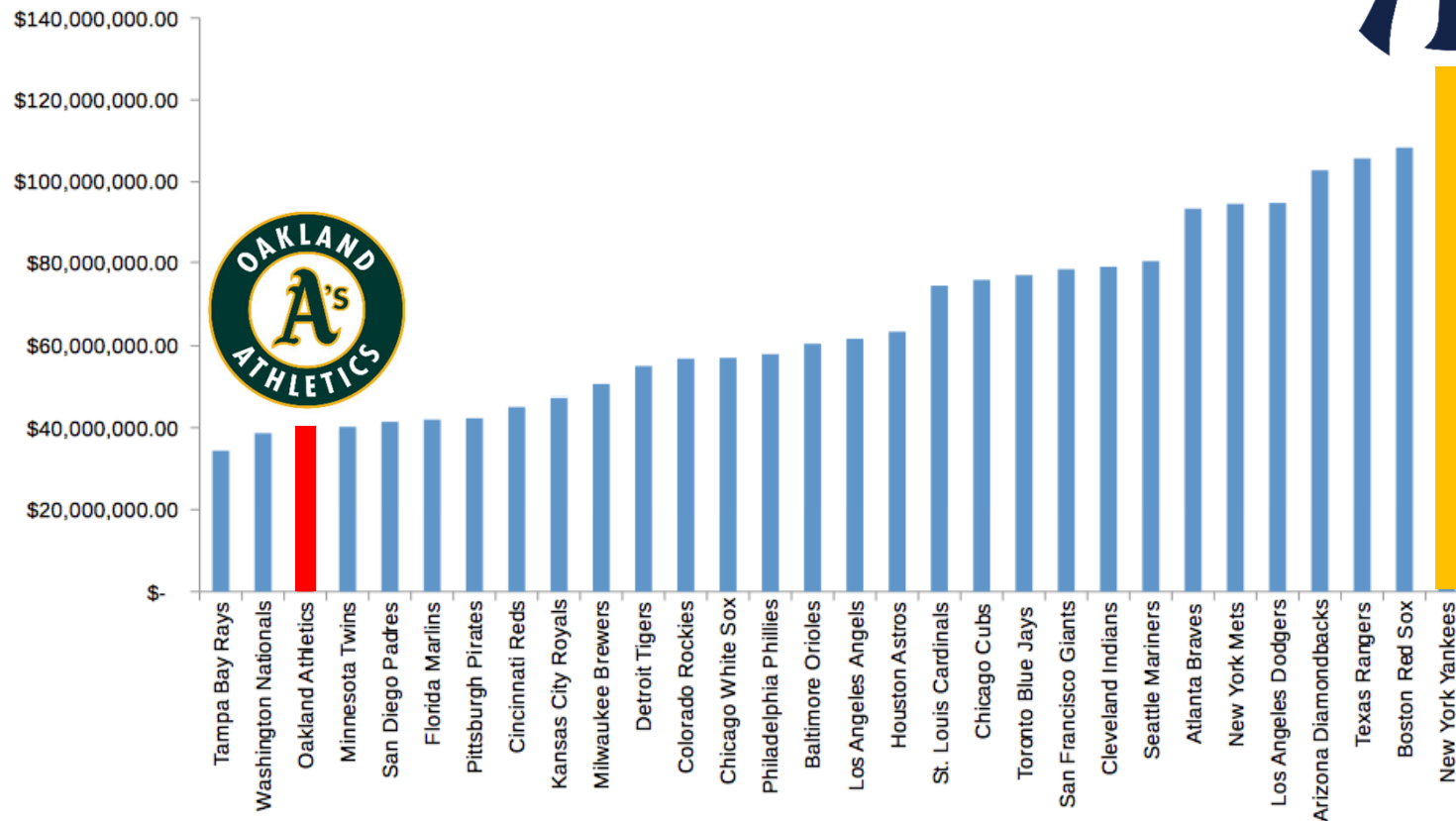
「野球における勝利の要因」

データを重視して選手をスカウト



チームの総年俸@MLB(2002)

この年、アスレックスは全球団最多の103勝!!!



NYヤンkeesはアスレックスの3倍!!!

セイバーメトリクス

野球において
データを統計学的見地から客観的に分析し、
選手の評価や戦略を考える分析手法

元ヤクルト野村監督のID野球もデータ重視



野球ライターで
野球史研究家・野球統計の専門家
ビル・ジェームズが1970年代に提唱

打者の成績を示す基準

□ 足の速さ

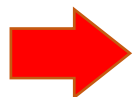
□ 打数 = 打席数 - (四死球 + 犠打 + 犠飛 + α)

□ **打率** = 安打 ÷ 打数

□ **打点** : 安打、犠打、犠飛、内野ゴロ、野手選択によって走者が得点した数

□ **出塁率** = (安打 + 四球 + 死球) ÷ (打数 + 四球 + 死球 + 犠飛)

□ **長打率** = 塁打 ÷ 打数



出塁率 × 3 + 長打率

マネーボール(2011)



○従来の野球スカウト
⇒経験と勘!!

○ビリー・ビーン
アスレチックスGM
ポール・デポDESTA

「野球における勝利の要因」
データを重視して選手をスカウト

➡ 現在NFL(アメフト)ブラウンズに
フロント入り!!
違う分野でも活躍できる!!!

野球でデータマイニング



① データの分類

過去の球団のあらゆるデータを分類・解析

② データの関連性を見つけ出す

チームの勝利に貢献する要素を見つけ出す

③ 事象の発生確率を予測する

活躍しそうな選手を予測する

→相場より安く獲得できる!!

映画の原作タイトル 「*Moneyball: The Art of Winning an Unfair Game*」



データマイニングの代表的な手法



(1) マーケットバスケット分析(相関ルール)

どの商品とどの商品を
どのような顧客が同時に購入したかを分析



店内の陳列方法を改善

amazon

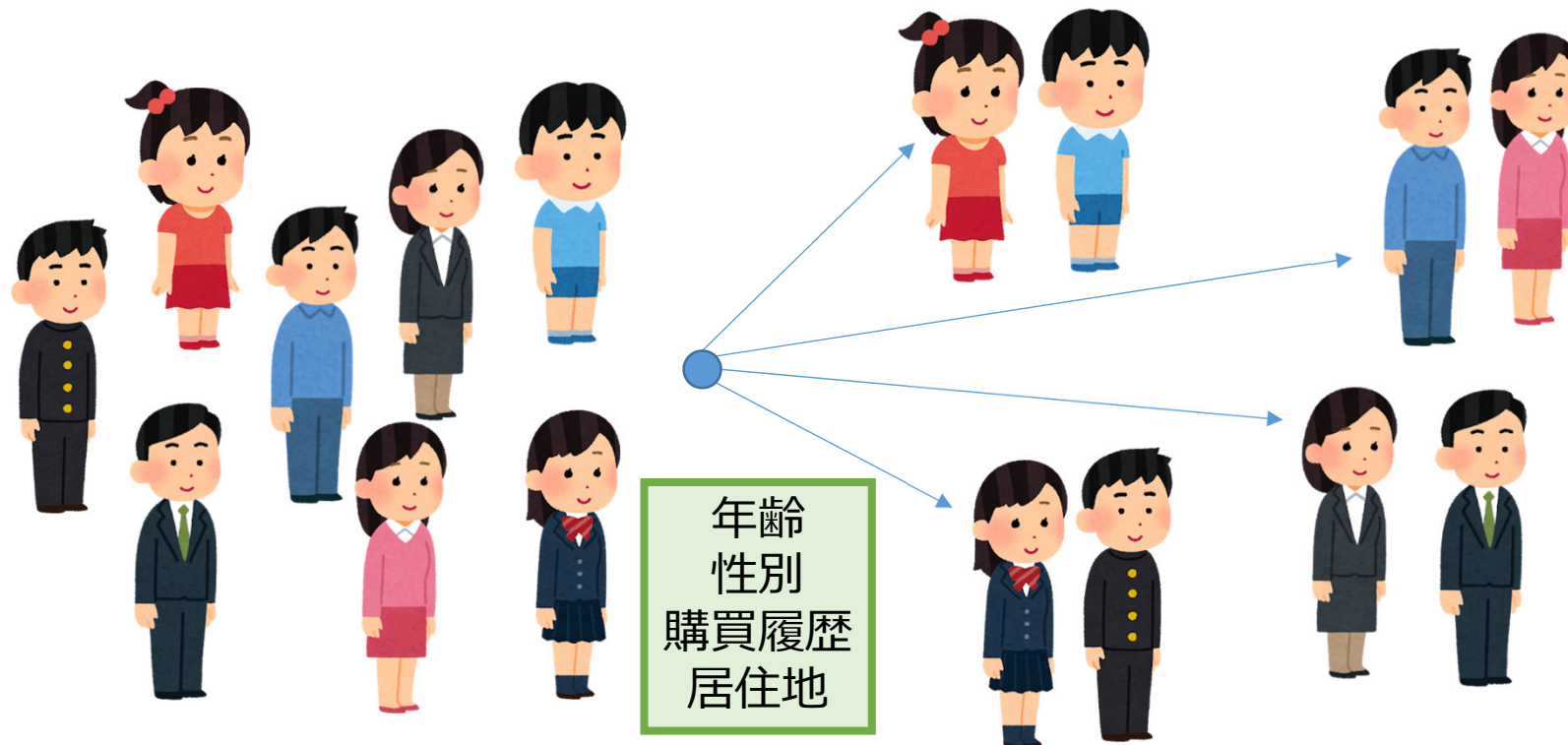
楽天
I C H I B A

この商品を買っている人は
これも買ってます

データマイニングの代表的な手法

(2) クラスタ分析

似ているデータごとにデータをまとめて分類
→適切な商品を推奨できる



データマイニングの代表的な手法



(3) ロジステック回帰分析

発生確率を予測する

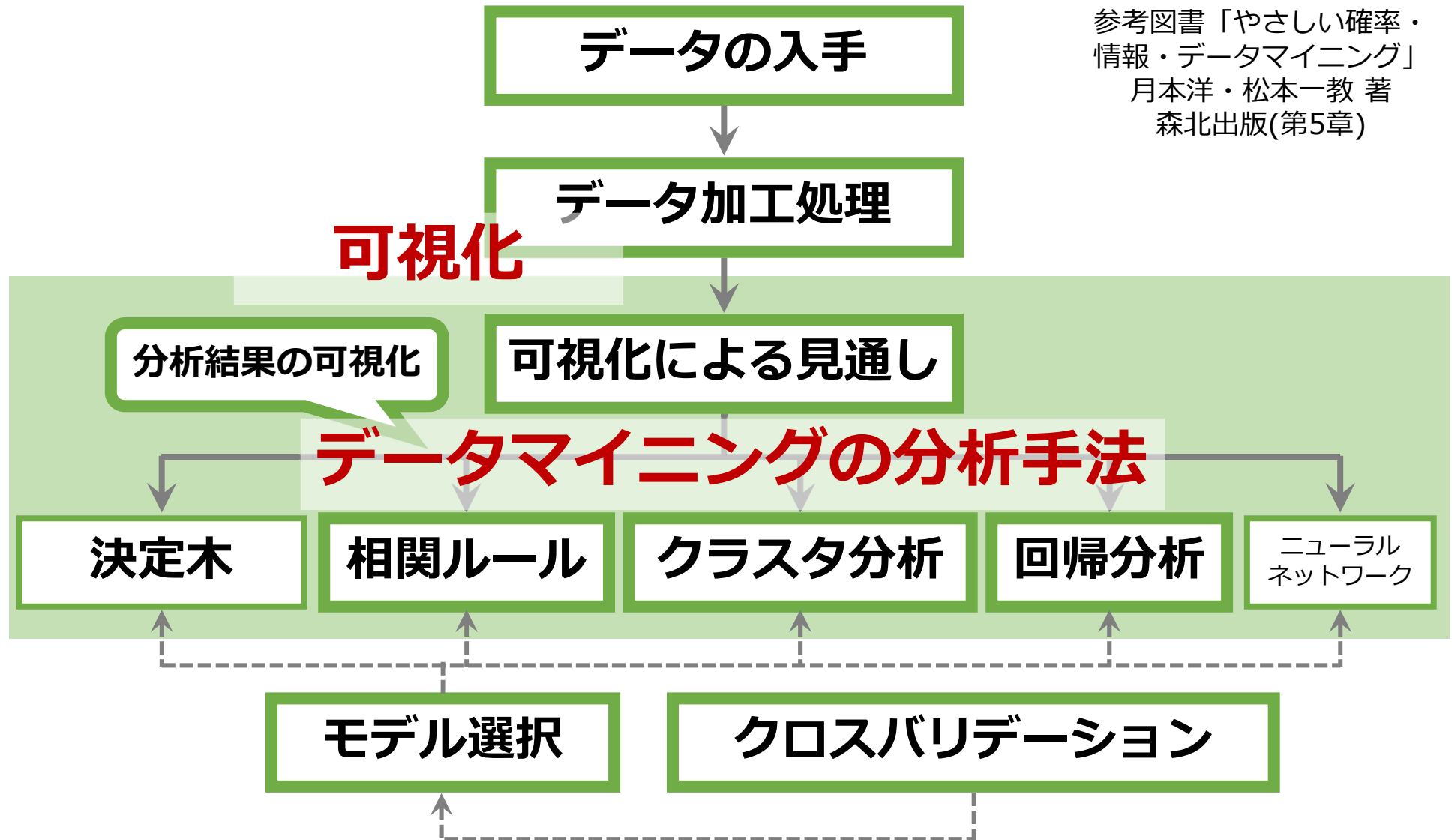
- がんの発症確率や生存率など
- アンケート結果から、携帯会社を乗り換える顧客を予測

 
  SoftBank

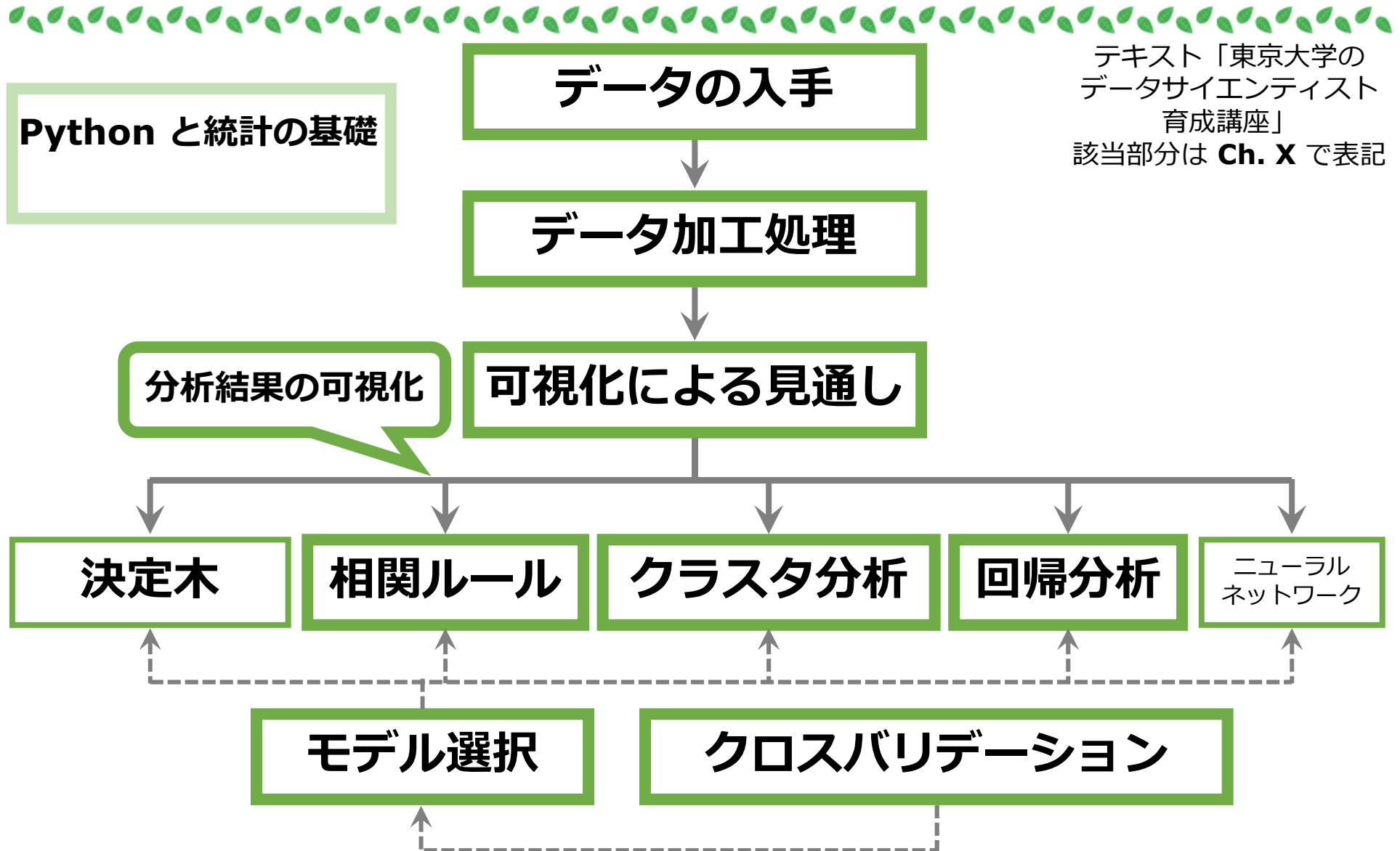


データマイニングの実際の手順

参考図書「やさしい確率・
情報・データマイニング」
月本洋・松本一教 著
森北出版(第5章)



データマイニングの実際の手順



授業計画



授業計画



授業計画



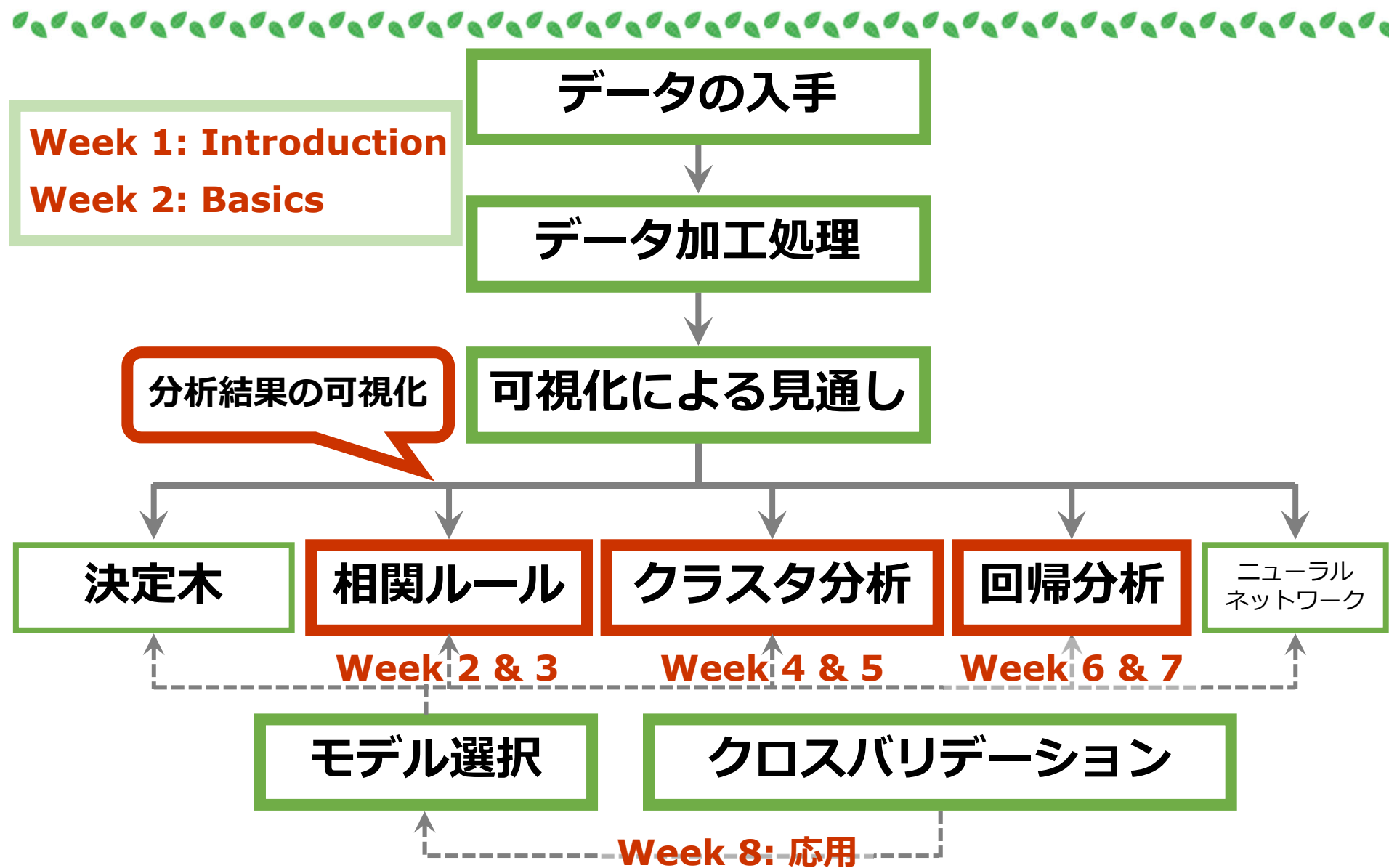
授業計画



授業計画




授業計画



授業計画



1. 概要と動作環境の確認
 2. マーケットバスケット分析
(講義と分析手法の基礎)
 3. マーケットバスケット分析
(サンプルデータで解析) & 可視化
 4. クラスタ分析 (講義と分析手法の基礎)
 5. クラスタ分析 (サンプルデータで解析) & 可視化
 6. 回帰分析 (講義と分析手法の基礎)
 7. 回帰分析 (サンプルデータで解析) & 可視化
 8. 応用・まとめ
- 

履修要件



- 基幹教育の理系ディシプリン
「プログラミング演習」の履修を前提とする。
Pythonを動かしたことがある。
→Yes?
→No?

 - PythonとJupyterの利用環境の整った
ノートPCを持参すること。
→PythonとJupyterの利用環境がない人は、
来週までに環境を整えてきてください。
Numpy と Matplotlib もインストール
-

履修要件



- 基幹教育の理系ディシプリン
「プログラミング演習」の履修を前提とする。
Pythonを動かしたことがある。
→Yes?
→No?

- **PythonとJupyterの利用環境の整った
ノートPCを持参すること。**
→PythonとJupyterの利用環境がない人は、
来週までに環境を整えてきてください。
Numpy と Matplotlib もインストール



この講義の目標



今後、データ解析をする必要ができたときに、
(調べながら)解析したいデータを

**自分でコードを書いて分析
結果を可視化**

できるようになること

 練習が必要



成績評価



□ 平常点(3点満点x8回分)


- 授業に出たうえでリフレクションシートを提出すると3点
- 3回欠席したら欠格

□ レポート(11点満点x7回分)

- おまけ問題には加点します。
- 3回以上課題の未提出があった場合
遅れて提出した課題は未提出0.5回分とし、
遅れた分の採点は2割減点とします。

□ 小テスト

今年は5週目に復習テストを行います。



遅刻・早退




授業は最初から最後まで出るのが基本

45分とか遅刻する人がそこそこいます。

演習の授業で大事な話はほとんど授業の前半でするので、「常識的な範囲内」を超えて遅刻したら欠席にします。

勝手に早退しちゃう人もいます。

演習の時間に課題でできなかったところを個別にフォローするので、勝手に早退した場合は欠席とみなします。



公欠（基幹教育履修要項p.18）

基幹教育院教務係で手続きを行ってください。

〔公認欠席に該当する事由〕	〔必要書類〕
① インフルエンザなどの感染症 （＊学校保健安全法施行規則第18条に規定する感染症）にかかった場合	診断書 ※ただし、インフルエンザに限っては、当面の間、状況に応じて添付書類を簡素化できますので基幹教育教務係に相談のこと
② 本学が、①の感染症にかかったおそれがあると認め、出校停止を指示した場合	
③ 2親等以内の親族が死亡した場合	会葬礼状 等の当該事由が確認できる書面
④ 裁判員候補者として裁判所に出向く場合及び裁判員として職務に従事する場合	選任手続き期間の通知（ 呼出状 ）等
⑤ 天災・交通機関の障害による場合	状況により 交通事業者の証明

公欠証明書を持ってきた時点で公欠とします。

講義資料・演習問題・宿題



OneDrive (九州大学Microsoft 365)で配布します。

Moodleのアナウンスにリンクがあります。

<https://office365.iii.kyushu-u.ac.jp/>
SSO-KIDでログインしてください。



講義資料・演習問題・宿題



Week1_JP_PPT.pdf : 講義資料日本語版
Week1_EN_PPT.pdf : 講義資料英語版
Ex_Week1.ipynb : 演習問題
Rep_Week1.ipynb : 課題 **(要提出)**

基本的には、上記のセットでファイルを毎週アップロードします。
必ずOneDriveからファイルを取得しておいてください。
課題を提出するときに、ファイル名を変える必要はありません。

教材

OneDriveから以下のファイルをDLしてください。

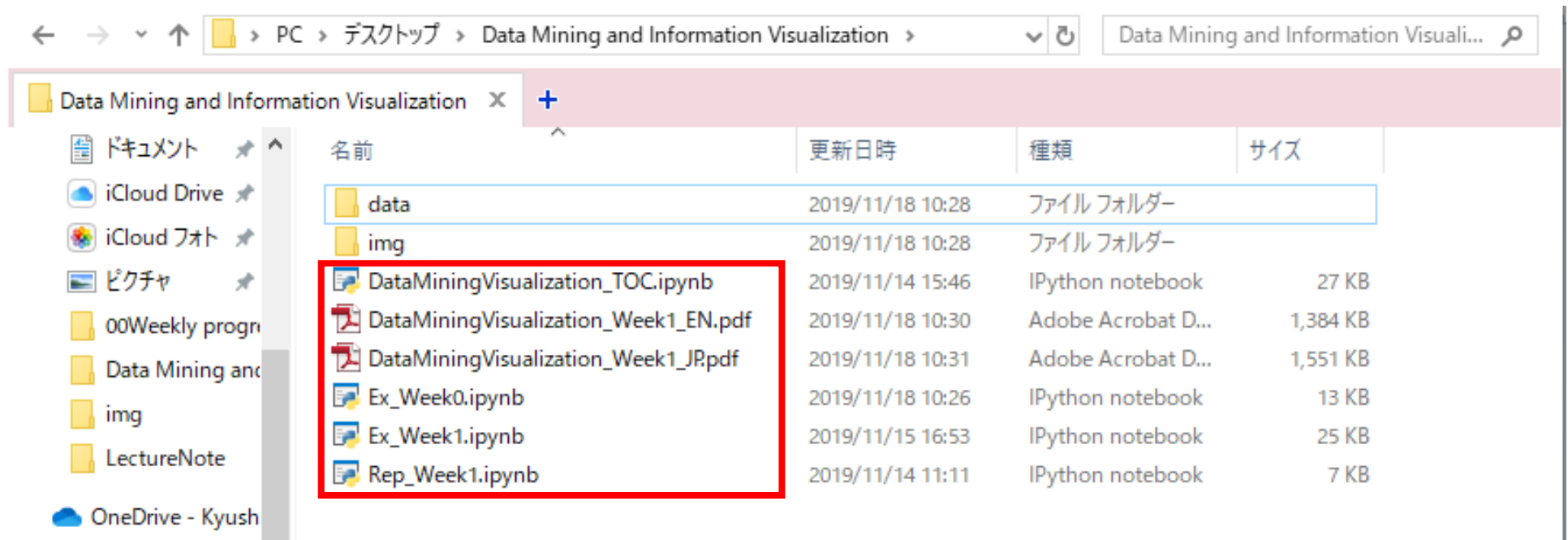
/Data Mining and Information Visualization/

というフォルダを作って、その中に、

.ipynb ファイル

.pdf ファイル

を保存してください。



教材

さらに。。。

/Data Mining and Information Visualization/

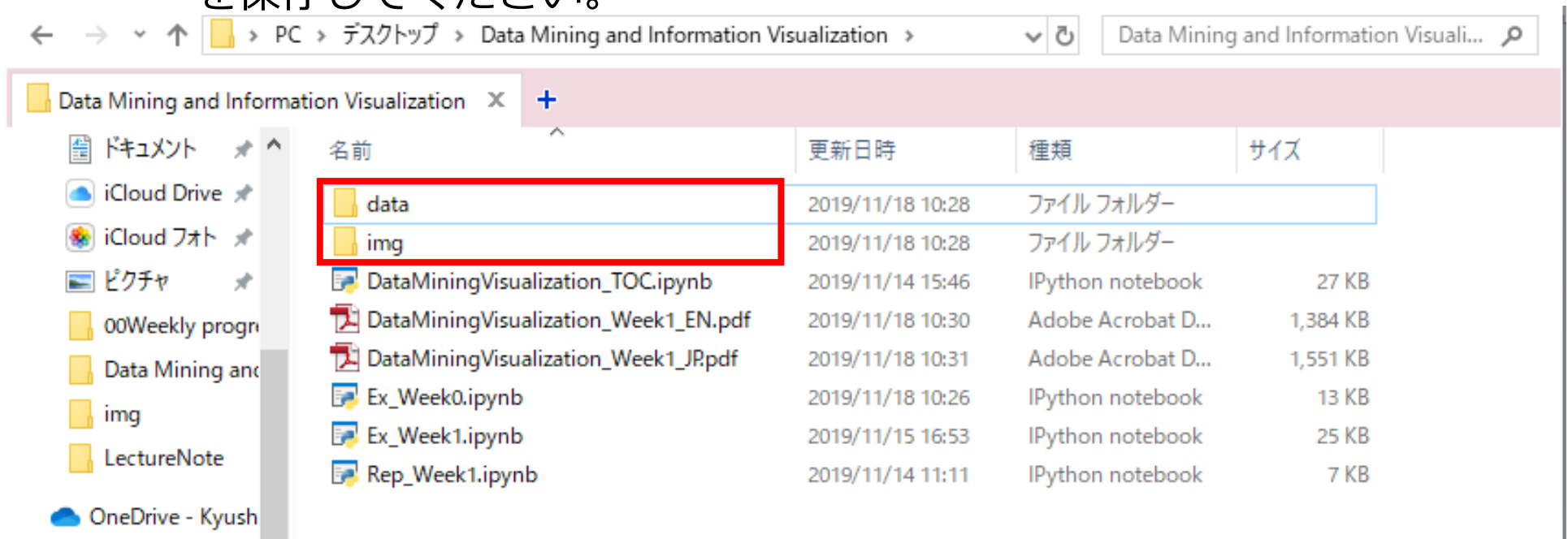
というフォルダの中に、「img」「data」フォルダを作って、

.csv ファイル

.txt ファイル

.xlsxファイル

を保存してください。



レポート



レポートは自分で書いてください。

- レポートは【必ず】自分で考えてください。
分からなければ、友達と相談しても構いません。
- プログラムは、**動く状態で提出**してください。
うまく動かない場合には、**どこで詰まったか、
説明文を付けて**ください。
- 自分で理解していない内容を写したレポートを
提出した場合は、不正行為となります。

→当該学期に履修した科目全ての単位を無効



レポート不正の例(1)(2019年度)



Markdown Python 3

Method 3-2: Regression (Analyze data & visualization)

レポート: 締め切り 2020年1月27日(月) / Assignment: Deadline Monday, January 27, 2020

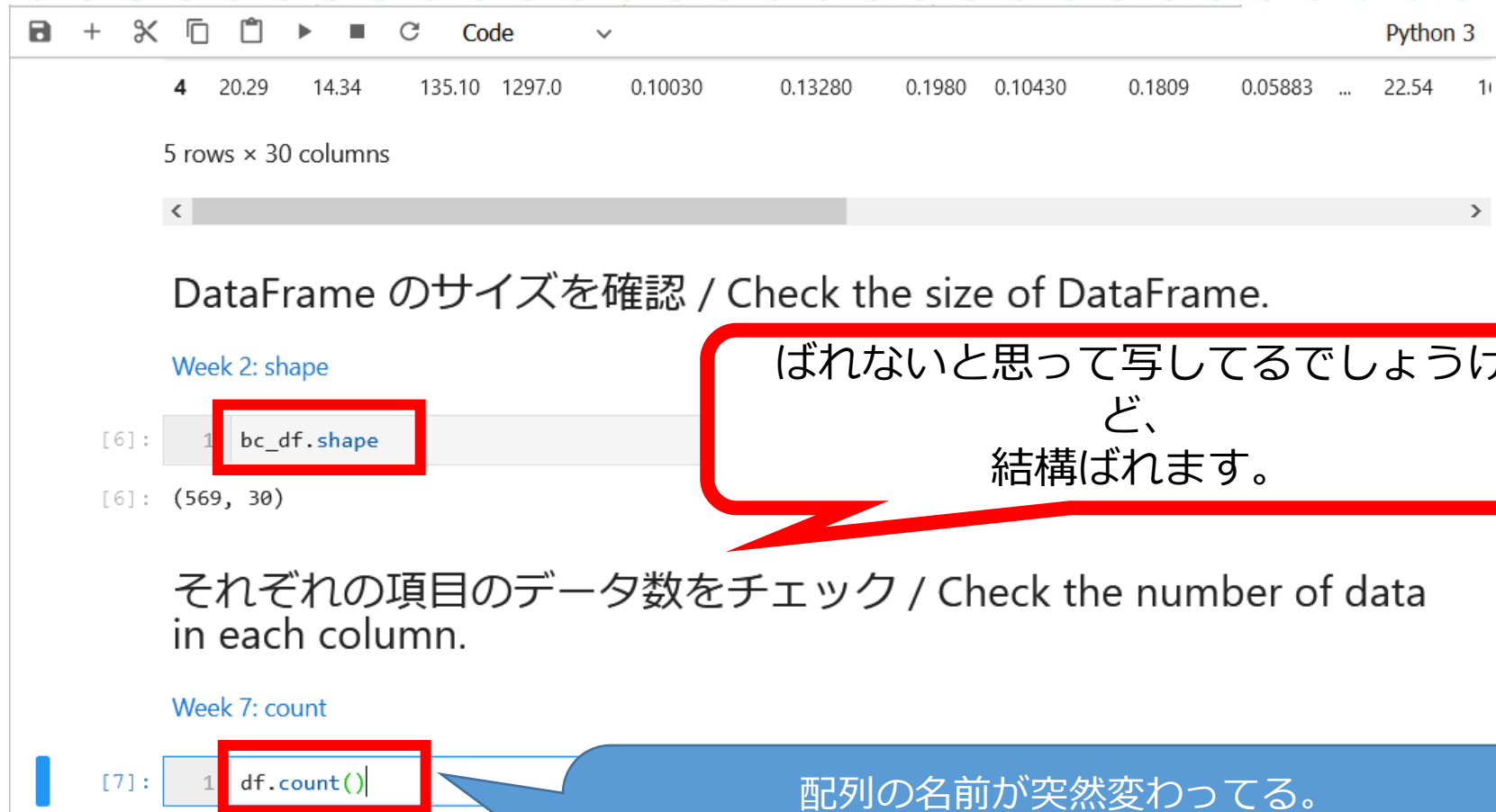
名前と学籍番号を表示してください。 / Please display your name and student ID.

[1]:
1 # print 関数を使って下さい / Use print function
2 print()

[2]:
1 # まずはライブラリをインポート / Import the libraries.
2 import numpy as np
3 import numpy.random as random
4 import scipy as sp
5 from pandas import Series, DataFrame
6 import pandas as pd
7

必要なライブラリがインポートされてない。
どうして動いたんでしょうね??

レポート不正の例(2)(2019年度)



4 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430 0.1809 0.05883 ... 22.54 1

5 rows × 30 columns

DataFrame のサイズを確認 / Check the size of DataFrame.

Week 2: shape

[6]: 1 bc_df.shape

[6]: (569, 30)

それぞれの項目のデータ数をチェック / Check the number of data in each column.

Week 7: count

[7]: 1 df.count()

ばれないと思って写してるでしょうけど、結構ばれます。

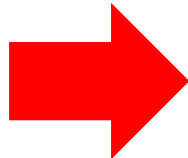
配列の名前が突然変わってる。
→丸写しはまずいと思って、配列の名前だけ変えたけど、途中で変えそびれているのがあった。

動かないはずなのに、どうして動いたんでしょうね??

レポート不正



悪質な場合には。。。

 基幹教育院に通報します。

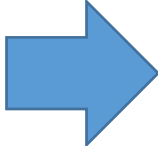
当該学期のすべての単位が
無効になります。



不正行為には厳正に対処します



正しいコードは一通りか??

 答えは「NO」

プログラミングスキルのレベルや人の性格がもろに出ます。
自分で書いていたら、一語一句誰かと同じコードには
絶対になりません。




レポート



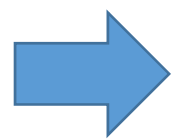
プログラムは、**動く状態で提出**してください。
うまく動かない場合には、どこで詰まったか、
【エラーをよく読んで、どこでうまく行かなくなっ
たか】 説明文を記入してください。

- 必要なライブラリがインポートできていない
 - 何らかの理由で途中からコードが実行できない
かつうまくいかなかった理由の記述がない場合
→点数はつけません
-

フィードバック

- 
- よかった点
 - 悪かった点
 - 分かりにくかった点
 - 直してほしい点。。。

皆さんからの感想を大募集してます。



**今年もリフレクションシートに
授業の感想などを毎週書いてください。
平常点の一部とします。**

(内容に対する点数ではなくて、提出したという行為自体に点が付きます。授業に関する苦情などで減点にはなりません。)

Preparation of Python libraries (1)

Install libraries with GUI of Anaconda

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Home

Environments

Learning

Community

Search Environments

base (root)

Shio EBATA

nodejs

python3

All

Channels

Update index...

matplotlib

X

Name

Description



basemap

Plot on map projections using matplotlib



descartes

Use geometric objects as matplotlib paths and patches.



matplotlib

Publication quality figures in python



mpl-scatter-density

Matplotlib helpers to make density scatter plots



mpld3

D3 viewer for matplotlib.

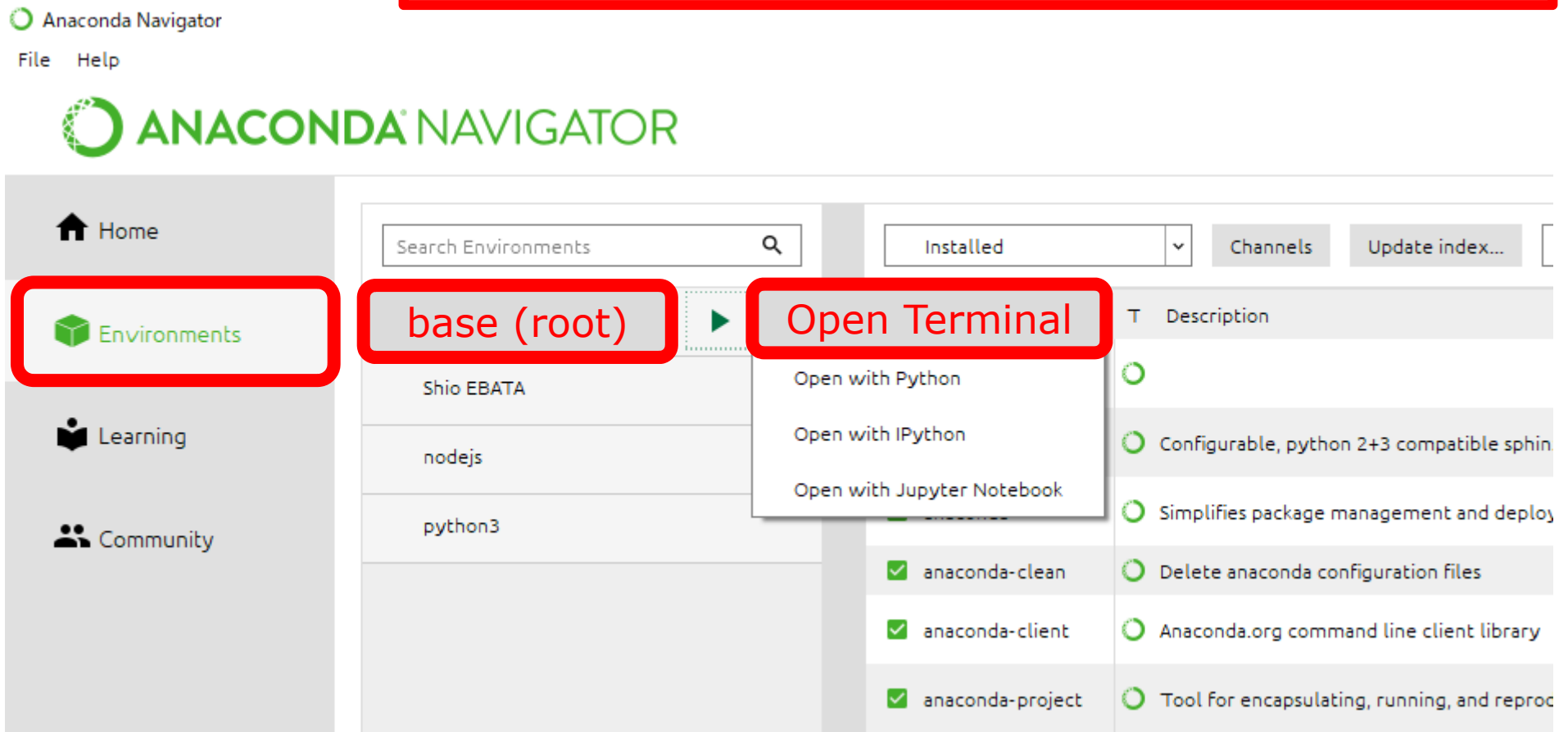
(1) "All" か "Not installed"を選択

(2) インストール
したいライブラリ
の名前を入力

(3) ボックスをチェック

Preparation of Python libraries (2)

Windows menu→start Anaconda Navigator



Preparation of Python libraries (2)



- メニューでEnvironmentを選択し、
仮想環境(Python3)を選んで
「Open Terminal」でターミナルを起動します。
- conda コマンドで各ライブラリをインストール

```
conda install -c conda-forge matplotlib  
conda install -c conda-forge numpy  
conda install -c conda-forge pandas  
conda install -c conda-forge scikit-learn
```

詳細は、Ex_Week0.ipynb を見てください。

Google Colaboratory



どうしても自分のパソコンで
Pythonが動かせない人は。。。。

Online Python environment
<https://colab.research.google.com/>



参考サイト : Jupyter Notebook



■ Jupyter Notebook を使ってみよう

<https://pythondatascience.plavox.info/python%E3%81%AE%E9%96%8B%E7%99%BA%E7%92%B0%E5%A2%83/jupyter-notebook%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6%E3%81%BF%E3%82%88%E3%81%86>



■ Jupyter Notebookのインストール方法や使い方

データ分析で欠かせない！ Jupyter Notebookの使い方 【初心者向け】

<https://techacademy.jp/magazine/17430>

■ Anaconda のインストールから jupyter notebook の起動まで

https://datachemeng.com/anaconda_jupyternotebook_install/



Jupyter Notebook



- 実行 Shift-Enter
- Ctr + / （スラッシュ）→複数行コメントアウト/解除
- 現在の出力をクリア
Cell→Current outputs→Clear

If文やforループは閉じないかわりに、
インデントで構造を決めるので、**インデントの場所に注意**

- Tab インデント(字下げ)
- Shift-Tab まとめて逆インデント(字上げ)

https://qiita.com/masafumi_miya/items/6524dbd227705351a00c

Jupyter Notebook



よく使うショートカットキー

- `esc+M` → マークダウンモード
- `esc+Y` → コードモード
- `esc+L` → 行番号を表示
- `esc+A` → セルを上挿入
- `esc+B` → セルを下挿入
- `esc+DD` → セルを削除

テキスト10ページ

もしくは

https://qiita.com/masafumi_miya/items/6524dbd227705351a00c

Week1 演習



- **Ex_Week0.ipynb**
Jupyter Notebook の基礎など
- **Ex_Week0_Pandas.ipynb**
Pandasの基礎
- **Ex_Week1.ipynb**
PythonとJupyter Notebook の基礎

提出は求めませんが、これらの演習問題の内容はできるようでない、今後の授業は難しいと思います。ぜひ一度目を通しておいてください。

ライブラリ = 工具箱 / Library = Toolbox

工具箱にはいろいろな道具が入っている。

In the toolbox, there are many tools.



工具 = 関数 / Tool = function

工具箱/Toolbox



Numpy

とんかち
のこぎり

ねじ Hammer
Saw
Bolt

道具の名前
Names of tools

工具箱の名前
Name of the toolbox

np.array 配列
np.dot 行列の積
np.identity 単位行列

関数の名前
Names of functions

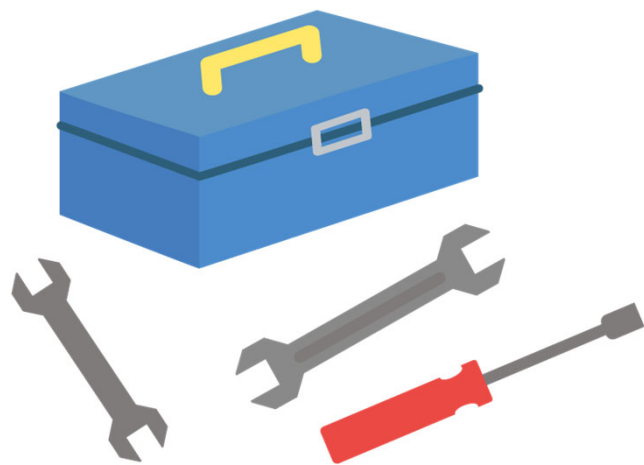
import numpy as np

Numpyという工具箱をnpという略称で読み込む

import=輸入する→読み込む、**使うと宣言する**(declare that you will use Numpy)

工具 = 関数 / Tool = function

工具箱/Toolbox



matplotlib.pyplot

とんかち
のこぎり
ねじ

道具の名前
Names of tools

工具箱の名前
Name of the toolbox

plt.plot

折れ線グラフ
Line graph

plt.scatter

散布図
Scattering plot

plt.bar

棒グラフ
Bar graph

関数の名前
Names of functions

```
import matplotlib.pyplot as plt
```

matplotlib.pyplotという工具箱をpltという略称で読み込む

import=輸入する→読み込む、**使うと宣言する**

(declare that you will use Matplotlib)

工具箱は何種類もある

Python

- print
- type

工具箱を宣言せずに使うことができる関数もある。

これらはPythonという一番大きな工具箱に入っている道具ともいえる。

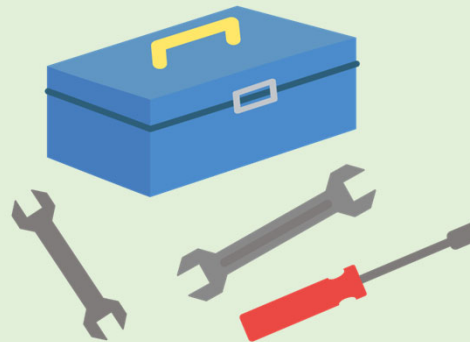
他にもまだまだたくさんある。
Pythonが便利なゆえんでもある。



Numpy

数値計算

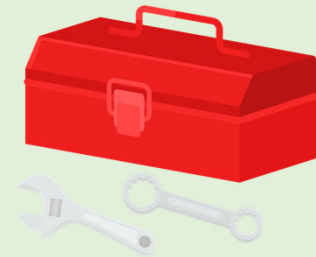
Numerical
calculation



matplotlib.pyplot

描画

Visualization



Scikit-learn

機械学習

Machine learning

Many kinds of (free) toolboxes

Python

There are many more, which is why Python is so useful.

- print
- type

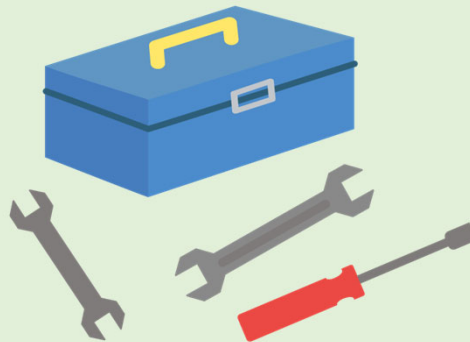
Some functions can be used in Python without declaring them. It can be said that such functions are in the biggest toolbox named Python.



Numpy

数值計算

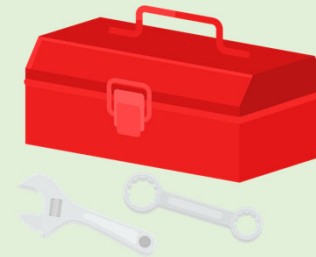
Numerical
calculation



matplotlib.pyplot

描画

Visualization



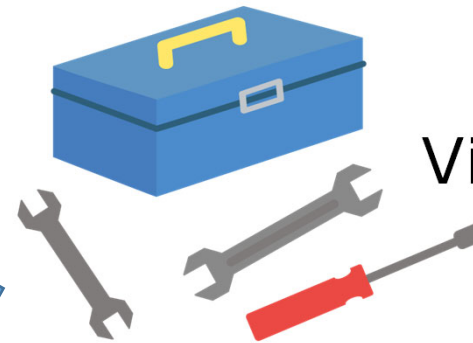
Scikit-learn

機械学習

Machine learning

import **matplotlib.pyplot** as plt

matplotlib



pyplot
描画

Visualization

Matplotlibという工具箱の中に
pyplotという工具箱が
入れ子構造に入っている

Inside the toolbox called
Matplotlib, a toolbox called pyplot
nested in the Matplotlib toolbox.

本当はほかにもMatplotlibの中には工具箱があって、pyplotのみ使います、と宣言している。
There are other toolboxes in Matplotlib, and we declare that we will only use pyplot.

引数 / Argument



道具(=関数)に渡す入力

The values you pass into the function

横軸を x、縦軸を y でプロットする

Plot x on the horizontal axis and y on the vertical axis.

```
import matplotlib.pyplot as plt  
plt.plot(x, y)
```

連番や等差数列を生成する

produce consecutive numbers and arithmetical progression

```
import numpy as np  
np.arange(0, 100, 2)
```

セルの実行



セルにコードを書いただけだと、
パソコンはその内容を認識してくれない。
セルに書いた計算結果を知りたいければ、
「セルを実行」 する必要がある。



便利なショートカット



Useful shortcuts



実行 / Run : **SHIFT-Enter**



上書き保存 / Save : **Ctrl-s**



コピー / Copy : **Ctrl-c**



貼り付け / Paste : **Ctrl-v**



切り取り/ Cut : **Ctrl-x**



コメントアウト / Comment out



CTRL-/
複数行が一括で
コメントアウトできる
Multiple lines can be
commented out at once.

CTRL-/
もう一回コメントアウトが除去
One more press removes comment out

新しいセルを作る

Insert a new cell



Escを一回押してから b を1回押す
Press Esc once, then press b.

セルを削除 / Delete a cell



Escを一回押してからdを2回押す
Press Esc once, then press d twice.

元に戻す/ Undo



Escを一回押してからzを1回押す
Press Esc once, then press z once.

おまけ：Matplotlibを試してみる



Jupyter Notebookで、
実際にグラフを描いてみよう!!

Ex_S_Matplotlib.ipynb



おまけ：グラフの軸を日本語化



- メニューでEnvironmentを選択し、仮想環境(Python3)を選んで「Open Terminal」でターミナルを起動します。
- pip コマンドで日本語化ライブラリをインストール

```
pip install japanize-matplotlib
```

日本語をラベルやタイトルに使いたいときは、
以下のようにインポートして使う。

```
import japanize_matplotlib
```

参考サイト <https://yolo.love/matplotlib/japanese/>

